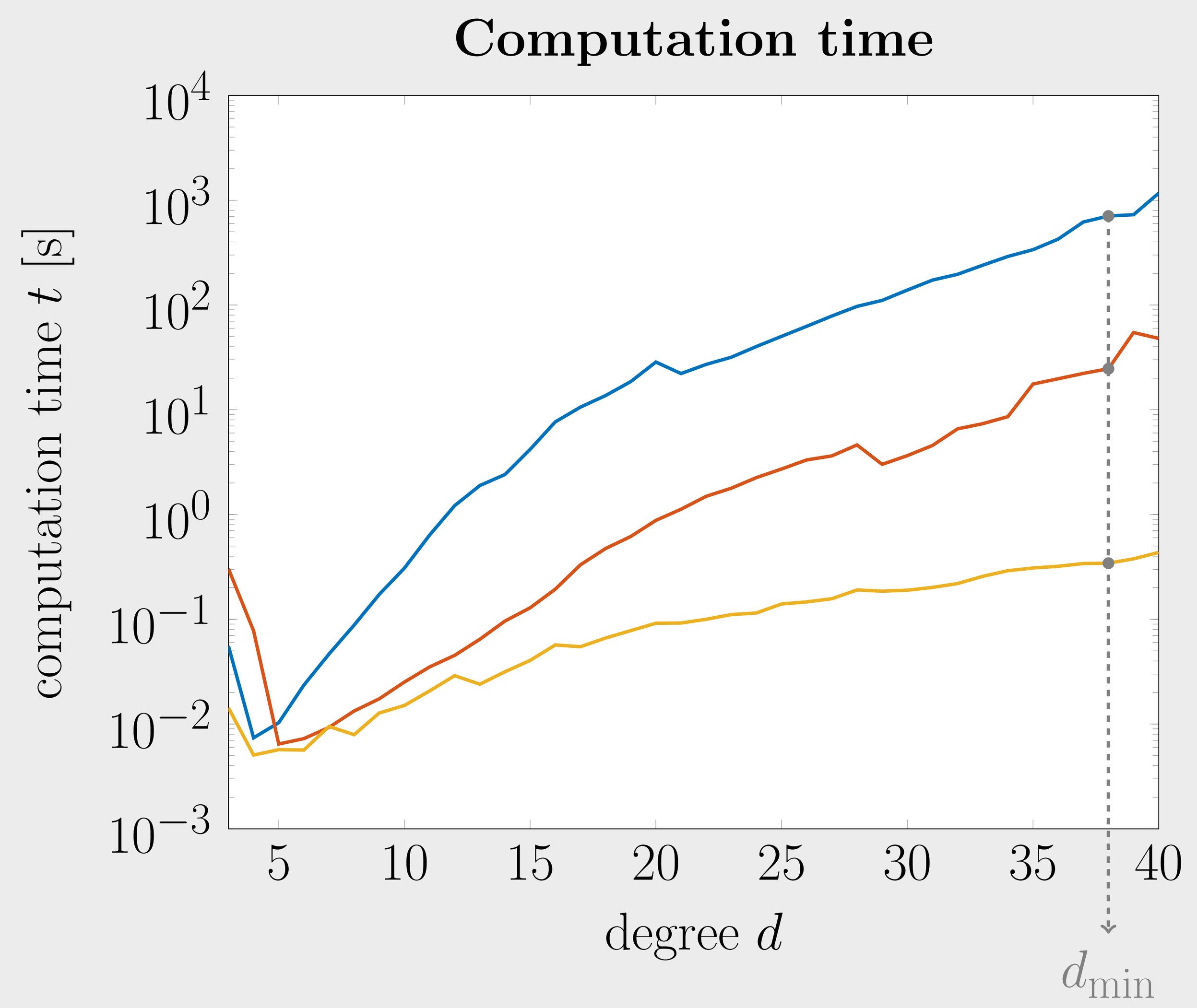
A Recursive Algorithm to Compute a Numerical Basis of the Null Space of the Block Macaulay Matrix

Poster for the 29th ERNSI Workshop in System Identification

Christof Vermeersch[†] and Bart De Moor[†]

A recursive and sparse algorithm pushes the block Macaulay matrix to larger problems!



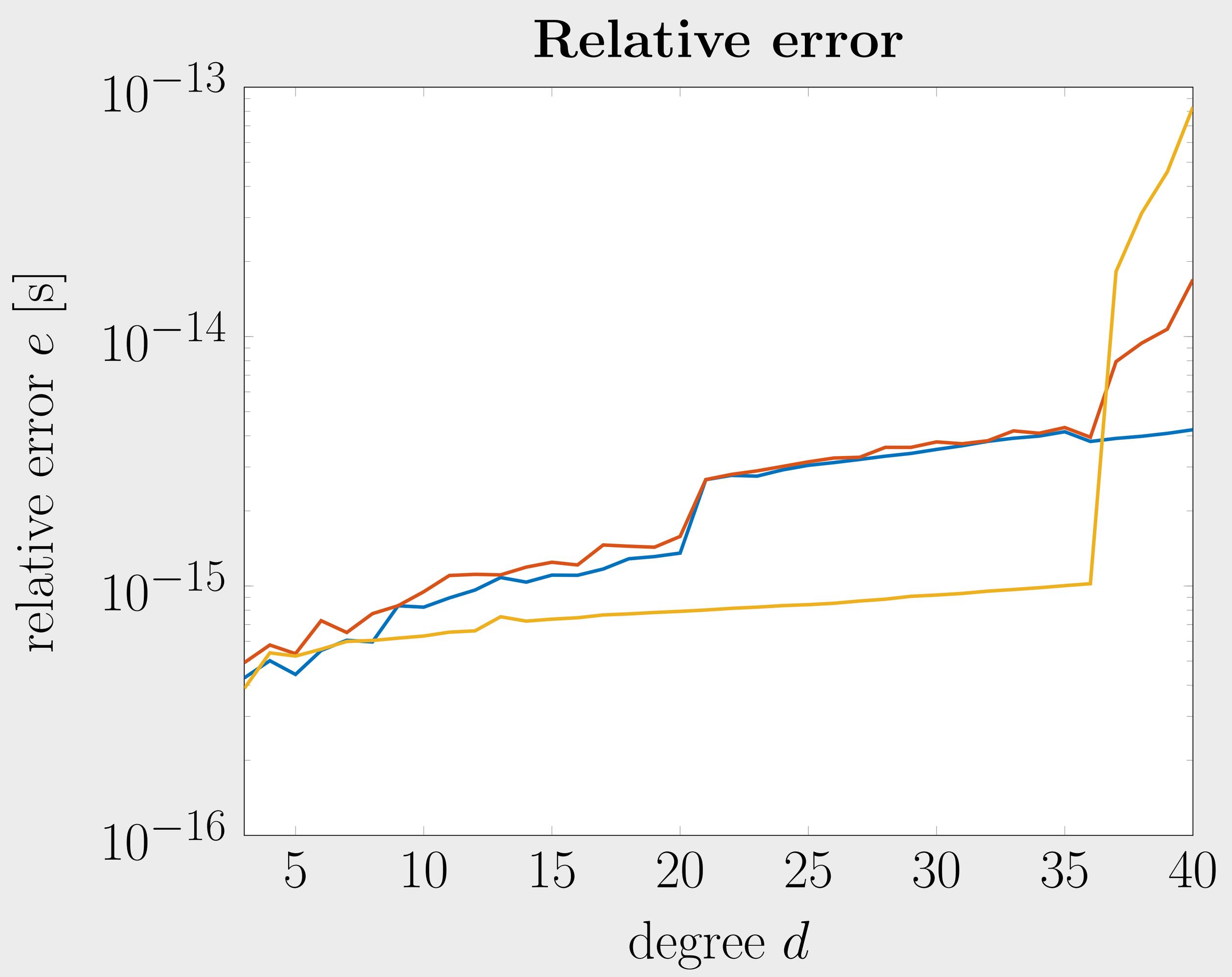


Figure 1: The computation time t and relative error $e = \frac{\|M_d Z_d\|}{\|M_d\|}$ of the full (–), recursive (–), and sparse (–) algorithm to compute a numerical null space of the block Macaulay matrix of the ARMA model identification example

[†] KU Leuven, Department of Electrical Engineering (ESAT), Center for Dynamical Systems, Signal Processing, and Data Analytics (STADIUS), Kasteelpark Arenberg 10, 3001 Leuven, Belgium (`christof.vermeersch@esat.kuleuven.be`, `bart.demoor@esat.kuleuven.be`)

Multiparameter Eigenvalue Problems

A multiparameter eigenvalue problem (MEP) extends the typical structure of the well-known standard eigenvalue problem (SEP)

$$\left(\sum_{\{\omega\}} A_\omega \lambda^\omega\right) z = 0, \quad z \neq 0$$

Examples:

- Linear 2-EP: $(A_{00} + A_{10}\lambda_1 + A_{01}\lambda_2)\, z = 0$
- Nonlinear 3-EP: $\left(A_{000} + A_{200}\lambda_1^2 + A_{013}\lambda_2\lambda_3^3\right) z = 0$

Block Macaulay Approach

We shift the MEP with the different eigenvalues and construct the block Macaulay matrix from the shifted coefficient matrices

$$\left(A_{00} + A_{10}\alpha + A_{01}\gamma + A_{20}\alpha^2 + A_{11}\alpha\gamma + A_{02}\gamma^2\right) z = 0$$

$$\begin{matrix} \times 1 \rightarrow \\ \times\alpha \rightarrow \\ \times\gamma \rightarrow \\ \vdots \end{matrix} \begin{bmatrix} A_{00} & A_{10} & A_{01} & A_{20} & A_{11} & A_{02} & 0 & 0 & \cdots \\ 0 & A_{00} & 0 & A_{10} & A_{01} & 0 & A_{20} & A_{11} & \cdots \\ 0 & 0 & A_{00} & 0 & A_{10} & A_{01} & 0 & A_{20} & \cdots \\ 0 & 0 & 0 & A_{00} & 0 & 0 & A_{10} & A_{01} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} z \\ \hline z\alpha \\ z\gamma \\ \hline z\alpha^2 \\ z\alpha\gamma \\ z\gamma^2 \\ \hline \vdots \end{bmatrix}$$

The null space of the block Macaulay matrix is **block multi-shift-invariant**

$$\begin{bmatrix} z \\ \hline z\alpha \\ z\gamma \\ \hline z\alpha^2 \\ z\alpha\gamma \\ z\gamma^2 \end{bmatrix} \xrightarrow{\alpha} \begin{bmatrix} z \\ \hline z\alpha \\ z\gamma \\ \hline z\alpha^2 \\ z\alpha\gamma \\ z\gamma^2 \end{bmatrix} \qquad S_1 K D_\alpha = S_\alpha K$$

(SEP)

In practice, we need to use a **numerical basis** of this null space

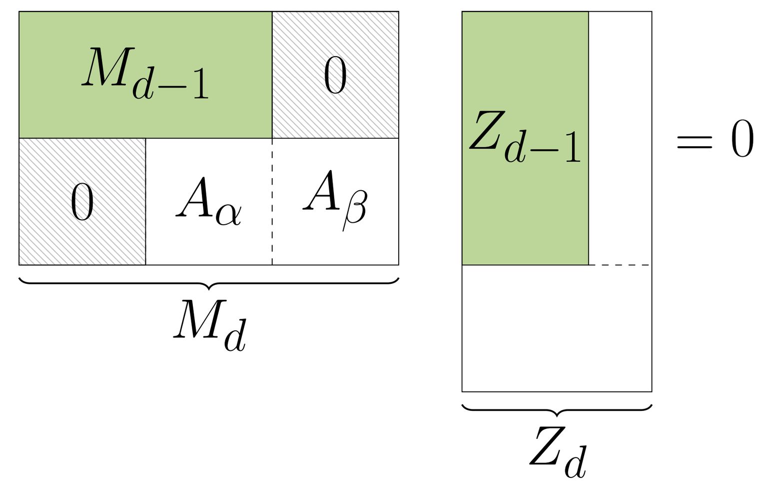
$$S_1 Z T D_\alpha = S_\alpha Z T$$
$$T D_\alpha T^{-1} = (S_1 Z)^\dagger (S_\alpha Z)$$

Problem statement

The computation of the numerical basis is a bottle neck, both in computation time and memory!

Recursive and Sparse Algorithm

The required degree d of the numerical basis Z_d is not known in advance, hence we need to build the block Macaulay matrix M_d **recursively** and update Z_d in every step



1. Enlarge the block Macaulay matrix M_{d-1} to M_d
2. Compute the update V_d from the new blocks A_α and A_β

$$V_d = \text{NULL}\left(\begin{bmatrix} A_\alpha Z_{d-1}^2 & A_\beta \end{bmatrix}\right)$$

3. Build a new numerical basis Z_d via the update V_d

$$Z_d = \begin{bmatrix} Z_{d-1} V_d^1 \\ V_d^2 \end{bmatrix}$$

A **sparse** implementation avoids the explicit construction of the block Macaulay matrix M_d and only stores the coefficient matrices

Numerical Example

We consider the identification of an ARMA model via an MEP, as discussed by Vermeersch and De Moor (2019)

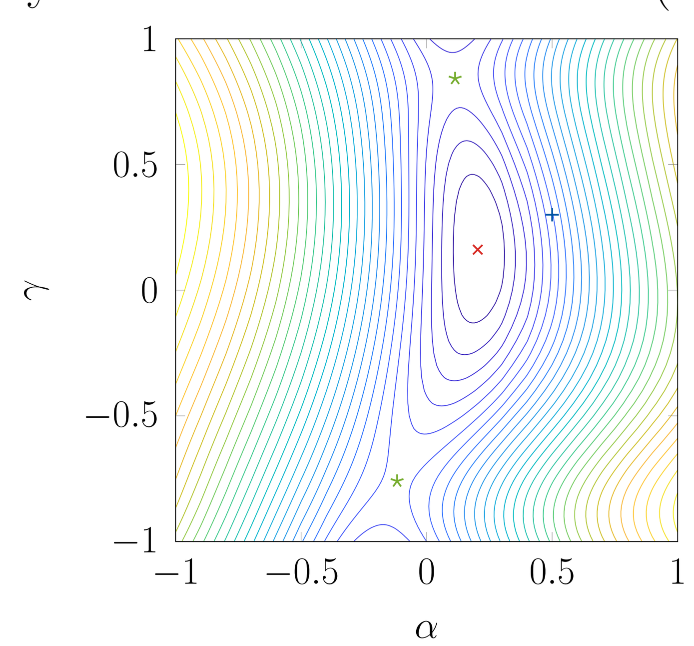


Figure 2: This contour plot shows the cost function of the identification problem, with the original, minimal and local parameters

Table 1: Numerical results of the identification problem ($d = 38$)

Algorithm	Time	Memory	Relative error
full orth.	3737.4s	2.1 GB	3.9060×10^{-15}
recursive orth.	144.1s	2.1 GB	1.8234×10^{-14}
sparse orth.	4.3s	0.1 GB	1.8234×10^{-14}